

GENERADORES AUTOMÁTICOS Y GRAMÁTICA FUNCIONAL INGLESA: UNA EXPERIENCIA DE INTEGRACIÓN EN EL AULA VIRTUAL

Julia Lavid López

lavid@filol.ucm.es

Juan Rafael Zamorano Mansilla

juanrafaelzm@yahoo.es

Facultad de Filología - UCM

Palabras clave: Generación automática; Gramática inglesa; Aula virtual

El propósito de este trabajo es describir una experiencia de integración en el aula virtual de un sistema computacional de desarrollo de gramáticas para la generación automática (Komet Penman Multilingual, KPML), típicamente utilizado para la investigación, como base para la exploración dinámica de la gramática funcional inglesa. La experiencia se desarrolló como parte de las actividades didácticas de la asignatura de Lingüística Computacional dentro del plan de estudios de Filología Inglesa, y que actualmente se ofrece como parte de la oferta de asignaturas virtualizadas en el Campus Virtual UCM.

1. INTRODUCCIÓN

El desarrollo de gramáticas computacionales con diferentes fines y aplicaciones es un campo muy activo de trabajo en la actualidad, tanto desde el punto de vista de la investigación lingüística como desde el ámbito de las aplicaciones comerciales. Algunas de las tareas para las que se han utilizado tradicionalmente las gramáticas computacionales han sido el análisis sintáctico, la traducción automática y la generación del lenguaje natural (GLN). Esta última puede definirse como aquella área de la Lingüística Computacional que se ocupa de la construcción de sistemas informáticos capaces de producir textos a partir de una representación no-lingüística de la información. Una definición más concisa de este campo la ofreció uno de sus más importantes investigadores, D. McDonald (1992), quien la definió como «el proceso de

construcción de textos con el fin de satisfacer propósitos comunicativos». Entre las motivaciones más importantes para la generación destaca principalmente una de carácter teórico, como es la creación de modelos psicolingüísticos de producción del lenguaje, y otra de carácter práctico, y que constituye uno de los principales objetivos para su desarrollo, es decir, la construcción de sistemas informáticos que produzcan textos coherentes en una o más lenguas. Así, en la actualidad existen sistemas que generan automáticamente informes meteorológicos (Kittredge *et al*, 1986), cartas en respuesta a peticiones de clientes (Springer *et al*, 1991; Coch *et al*, 1995), programas para la producción de documentación técnica (Reiter *et al*, 1995; Rösner y Stede, 1994), textos conteniendo instrucciones (Not y Stock, 1994; Lavid 1995, 1996; Paris *et al*, 1995), información y educación sanitaria del paciente (Cawsey *et al*,

1995; DiMarco *et al.*, 1995), informes médicos (Li *et al.*, 1986), e interfaces de lenguaje natural a bases de datos, entre otros.

En general, aunque la mayoría de las aplicaciones prácticas de la GLN se ha centrado en el monólogo, el ámbito de aplicaciones se ha ampliado también al diálogo, siendo actualmente un componente importante en sistemas tutoriales para la enseñanza asistida por ordenador, sistemas de ayuda o de explicación de sistemas expertos, o en sistemas de información y reserva de viajes, entre otros (Lavid, en prensa)¹.

A pesar de esta variedad de aplicaciones de la GLN, su potencial como herramienta pedagógica está aún por explotar, existiendo en la actualidad algunas propuestas en esta dirección (véase Zamorano, 2002).

Este trabajo describe una experiencia piloto de integración en el aula virtual de un sistema computacional de desarrollo de gramáticas para la generación automática (Komet Penman Multilingual, KPML), típicamente utilizado para la investigación, como base para la exploración dinámica de la gramática inglesa. La experiencia se desarrolló como parte de las actividades didácticas de la asignatura de Lingüística Computacional, optativa de segundo ciclo dentro del plan de estudios de Filología Inglesa, y fue programada como parte de las tareas de virtualización de dicha asignatura dentro de un proyecto de innovación docente de la Universidad Complutense (véase la sección de agradecimientos).

El trabajo se estructura como sigue: en la sección 2 se ofrece una descripción general de los sistemas de generación basados en rasgos, y se justifica la elección del generador KPML para la presente experiencia didáctica. La sección 3 explica cómo se planificó la experiencia y el procedimiento para su implementación en el aula virtual. La sección 4 presenta los resultados de la actividad realizada con los alumnos comentando los puntos más destacables desde el punto de vista didáctico. Finalmente la sección 5 resume los

puntos más destacables del trabajo y presenta unas conclusiones preliminares.

2. KPML Y LOS SISTEMAS BASADOS EN RASGOS

Los diferentes sistemas de generación existentes utilizan diversos métodos de generación, caracterizados por su menor o mayor grado de sofisticación: desde el método más simple basado en la técnica del texto enlatado hasta la generación basada en rasgos (*feature-based systems*). En estos sistemas cada rasgo representa una alternativa posible de expresión: por ejemplo, la oración que deseamos generar puede ser positiva o negativa, imperativa o indicativa, en tiempo presente o pasado. Según este método cada expresión lingüística puede caracterizarse por un conjunto de rasgos. Si deseamos generar una oración específica, el sistema recopilará los rasgos apropiados correspondientes a dicha oración hasta que esté completamente determinada.

La ventaja de la generación basada en rasgos radica en la simplicidad y elegancia en su concepción. El problema principal está en la dificultad de mantener interrelaciones de rasgos y control de su selección. Entre los sistemas más avanzados que utilizan este método destacan el sistema PENMAN (Mann, 1983) y su sucesor KPML (Bateman y Matthiessen, 1991; Bateman, 1997), COMMUNAL (Fawcett, 1990), FUF/Surge (Elhadad y Robin, 1997), y POPEL (Reiter, 1990). Mientras los tres primeros utilizan gramáticas sistémicas para el proceso de realización lingüística, los dos segundos utilizan gramáticas de unificación.

Para la experiencia docente que se describe en este trabajo se escogió el sistema KPML (Komet Penman Multilingual) por las siguientes razones:

1. El sistema forma parte de un entorno de desarrollo de gramáticas que lo hacen especialmente interesante para la exploración dinámica de los aspectos funcionales de la lengua.
2. Está equipado con herramientas gráficas que permiten estudiar diferentes re-

¹ Véase el capítulo 4 de Lavid (2005) para un desarrollo detallado de los diferentes componentes y procesos de generación así como los métodos principales.

giones de la gramática o de la estructura de la oración generada.

3. Contiene no sólo una gramática computacional de la lengua inglesa (Nigel), desarrollada en el proyecto Penman y basada en el modelo sistémico-funcional de Halliday (Mann y Matthiessen, 1985), siendo posible obtener también gramáticas completas o parcialmente desarrolladas para otras lenguas tales como el alemán, el holandés, el búlgaro y el español, algunas de ellas en proceso de construcción (véase Lavid *et al*, 2002; Lavid, 2003).

A continuación se describe la actividad realizada destacando, en primer lugar, la planificación y, en segundo lugar, los resultados de la actividad de generación conjunta con los alumnos.

trada para la generación, etc.) y cómo estos componentes se relacionan entre sí durante el proceso de generación. La figura 1 ilustra los principales componentes de este sistema:

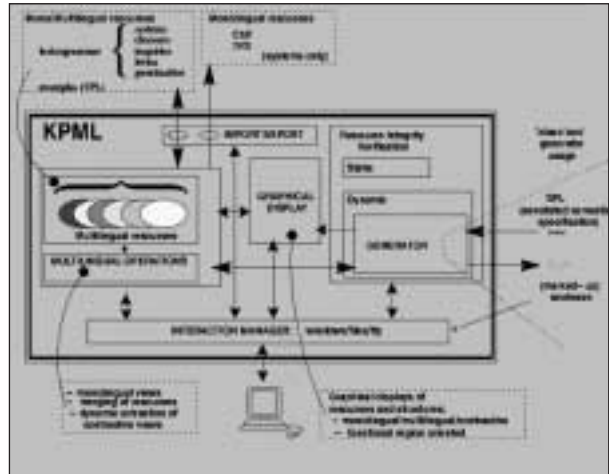


Figura 1. Componentes internos del sistema KPML.

3. PLANIFICACIÓN DE LA ACTIVIDAD

3.1. ASPECTOS PRELIMINARES

El propósito de la actividad planificada en el Campus Virtual fue ofrecer a los estudiantes la oportunidad de estudiar y observar la lengua en su aspecto dinámico, permitiéndoles interactuar de manera directa con la gramática de la lengua. La actividad fue integrada en la clase de la asignatura de Lingüística Computacional (lengua inglesa) como puesta en práctica de diversos aspectos teóricos sobre componentes y métodos de generación automática, así como de gramáticas funcionales adaptadas para la generación.

3.2. PROCEDIMIENTO

El procedimiento fue el siguiente: en primer lugar se hizo a los estudiantes una demostración en la que se mostraron los componentes fundamentales de KPML (las partes de una gramática computacional, los datos de en-

Esta demostración finalizó con la creación de manera conjunta con los alumnos del formalismo necesario para la generación de una oración simple que sirviera de ejemplo. Dichos formalismos reciben el nombre de SPL. La figura 2 ilustra un ejemplo de SPL para la oración *The dog swam across the river*.

```
(in-package "KPMML")

(in-language :languages :ENGLISH)

(EXAMPLE
 :NAME      place_prep-1
 :TARGETFORM "The dog swam across
the river."
 :LOGICALFORM
 (x / nondirected-action)
 :ACTOR
 (dog / dog :determiner the)

 :nonparallel-extent
 (nin / empty :lex river
determiner the)

 :LEX swim
 :tense past )

 :SET-NAME examples )
```

Figura 2. Ejemplo de SPL.

Dicho SPL, al ser introducido como datos de entrada en el generador y cruzarse con los recursos lingüísticos de la lengua inglesa, da lugar a la oración deseada. La figura 3 muestra el resultado generado por el programa, indicando mediante un diagrama de árbol la estructura de la oración:



Figura 3. Estructura de la oración generada automáticamente The dog swam across the river.

La actividad de los alumnos se centró en la creación de formalismos de SPL para la generación de diez oraciones con el sistema KPML. La razón fundamental para concentrarse en esta actividad fue que cada SPL constituye de hecho un análisis de tipo funcional de la oración que se pretende generar. Así, a la hora de diseñar un SPL los alumnos deben llevar a cabo un análisis preliminar de los rasgos semánticos funcionales de la oración que se pretende generar, tales como el tipo de proceso que representa la oración, los participantes involucrados en dicho proceso, las circunstancias asociadas, etc. Dicho análisis, además, está basado en los principios de la gramática funcional, teoría con la que los alumnos están bastante familiarizados por estar presente en un buen número de las asignaturas que oferta nuestro Departamento de Filología Inglesa.

A pesar del conocimiento lingüístico que poseen la mayoría de los participantes en el curso, fue necesaria la creación de un tutorial en línea en el que se explicaron los comandos necesarios para especificar los rasgos semántico-funcionales en los SPL objeto del ejercicio. Dicho tutorial se integró como parte de la unidad 3 del curso virtual de la asignatura de Lingüís-

tica Computacional, de forma que los alumnos pudieran consultarlo en línea y formular dudas y preguntas a los docentes mediante las herramientas de comunicaciones que ofrece la plataforma WebCT del Campus Virtual UCM.

Además, el tutorial en línea se dividió en diferentes secciones para facilitar la consulta por parte de los alumnos. La figura 4 muestra la sección introductoria de dicho tutorial en el que se definen los conceptos esenciales para la creación de formalismos SPL y los comandos principales para la generación de oraciones.



Figura 4. Sección introductoria del tutorial en línea.

Las principales secciones del tutorial son: la sección introductoria, la sección que explica las propiedades de las oraciones, y tres secciones explicativas de los principales tipos de procesos, participantes y circunstancias asociadas. La figura 5 ofrece una vista parcial de la sección explicativa de las propiedades oracionales:



Figura 5. Sección explicativa de las propiedades oracionales.

Por su parte, la figura 6 muestra la sección explicativa de los tipos de procesos necesarios para generar diferentes tipos de oraciones.



Figura 6. Sección explicativa de los tipos de procesos.

Además se creó un archivo con los ejercicios de generación de oraciones que tenían que preparar los alumnos, tal y como muestra la figura 7:



Figura 7. Archivo de ejercicios.

El archivo de ejercicios consistió en una lista de oraciones para las cuales debían diseñar los formalismos SPL correspondientes. Como ya se ha señalado anteriormente, la tarea de diseñar el SPL para la generación de una oración determinada equivale de hecho a realizar un análisis de dicha oración. Además, para mayor comodidad de los estudiantes, se les proporcionó un archivo con diez SPL —uno por cada oración que debían generar— con el fin de que

los alumnos sólo tuvieran que editar las partes relevantes para la actividad (tipo de proceso, participantes, etc.), y evitar así en lo posible los errores debidos a factores más relacionados con la computación.

4. DESCRIPCIÓN DE RESULTADOS

Después de dos semanas de trabajo en grupo con ayuda del material suministrado, los alumnos trajeron a clase sus formalismos SPL almacenados en un único archivo. Dicho archivo fue cargado en el sistema KPML con el propósito de generar a partir de las especificaciones que contenía y observar el resultado obtenido por los distintos grupos para cada oración de la lista.

Durante la sesión presencial se realizó una actividad de aprendizaje cooperativo comentando las generaciones resultantes con los estudiantes. En los casos en que dichas generaciones fueron las correctas, no fue necesario, por lo general, ningún trabajo o comentario adicional. Pero los casos más interesantes desde un punto de vista pedagógico fueron aquellos en los que la generación obtenida resultó incorrecta.

En ocasiones el problema se debió a meras causas técnicas: los alumnos, a pesar de las precauciones tomadas en este sentido, habían escrito de manera incorrecta un comando o habían hecho un uso erróneo de los símbolos propios del formalismo, en especial los paréntesis, los espacios o los dos puntos. Estos casos fueron resueltos en clase con la ayuda del profesor sin más comentario.

En otras ocasiones, sin embargo, el error en la generación era consecuencia directa de un análisis incorrecto por parte del alumno de la oración a generar. Fue entonces cuando el contraste entre la forma obtenida y la forma deseada resultó especialmente útil para explicar a los estudiantes cuál era el origen del error. Entre los errores más comunes que creemos contribuyeron a mejorar el conocimiento de los fundamentos de análisis lingüístico por parte de los estudiantes destacamos los siguientes:

- a) Un agrupamiento equivocado de los constituyentes a la hora de diseñar el

SPL de la oración a generar. Por ejemplo, cuando los alumnos intentaron obtener la frase *The weather in Madrid is sunny*, algunos se encontraron con el resultado *The weather is sunny in Madrid*. Al examinar la estructura resultante de la oración generada con KPML fue fácil hacer comprender a los alumnos que en la oración objetivo el constituyente *in Madrid* dependía del participante *The weather*, y no del conjunto de la oración, como habían hecho en su SPL los alumnos que obtuvieron una generación fallida. Las figuras 8 y 9 muestran los distintos SPL diseñados por los alumnos, donde se aprecia la diferente posición que dieron al constituyente *in Madrid* dentro de la oración.

```
:LOGICALFORM
(x / property-ascription
:domain (p1 / weather :determiner the)
:range (p2 / sunny)
:three-d-location (p3 / madrid)
)
```

Figura 8. SPL que genera la oración *The weather is sunny in Madrid*.

```
:LOGICALFORM
(x / property-ascription
:domain (p1 / weather :determiner the
:three-d-location (p3 / madrid)
)
:range (p2 / sunny)
)
```

Figura 9. SPL que genera la oración *The weather in Madrid is sunny*.

Y en las figuras 10 y 11 se observa la diferente estructura que genera KPML a partir de ambos SPL.

- b) Una especificación poco precisa acerca del tipo de proceso que aparecía en la oración o las características semánticas de alguno de los participantes. Por ejemplo, algunos alumnos clasificaron el proceso en *Mary forgot the names* como mental. Esta clasificación es



Figura 10. Estructura de *The weather in Madrid is sunny*.

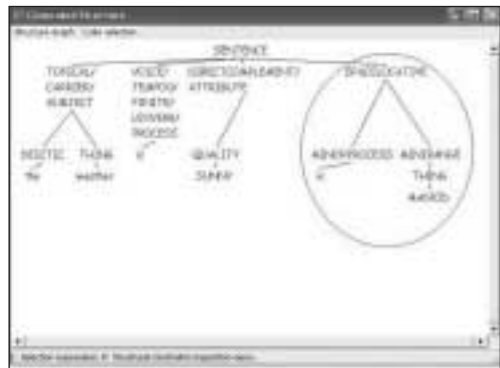


Figura 11. Estructura de *The weather is sunny in Madrid*.

acertada, desde luego, pero a la hora de generar, el programa requería una información más específica. Más concretamente, necesitaba saber qué tipo de proceso mental contenía la oración objetivo. Como consecuencia de esto, el proceso de generación era interrumpido y el generador ofrecía un menú a los estudiantes con las tres opciones posibles, dentro del marco de la gramática sistémico funcional, para un proceso mental: proceso de reacción, cognitivo o de percepción. El que los alumnos decidiesen correctamente entre estas tres opciones conduciría lógicamente a una generación correcta o incorrecta. Este hecho resultó de interés pedagógico por dos razones; en primer lugar reforzaba los conocimientos

teóricos de los alumnos acerca de la clasificación de los diferentes tipos de proceso según el modelo sistémico funcional, tratado en un buen número de las asignaturas de la carrera. En segundo lugar ofrecía a los estudiantes la oportunidad de involucrarse de manera directa y activa en el proceso de selección de rasgos de la gramática y la generación de un resultado. Esto supuso claramente un elemento adicional de motivación ausente en las clases en las que se practica el modo tradicional de análisis gramatical, en el que los alumnos tienen un papel mucho más pasivo. La figura 12 muestra el menú que el programa presentaba a los estudiantes.



Figura 12. Menú de elección en el sistema de tipo de proceso mental.

- c) Una especificación incorrecta de los participantes en relación con el tipo de proceso de la oración. Así, un grupo de alumnos definió el proceso en la oración *He told the truth* como verbal, lo cual es correcto. Sin embargo, definieron el participante *the truth* como *actee*, cuando este tipo de participante sólo se da en procesos materiales. La consecuencia es que el programa ignoraba dicho participante por ser inesperado en un proceso de tipo verbal, siendo el resultado gene-

rado *He told*. La generación correcta se obtuvo al reemplazar el término *actee* por *saying*, participante este último que sí va normalmente asociado a los procesos verbales.

- d) Debido a las características de las gramáticas de generación, es posible obtener el resultado deseado a partir de más de una especificación semántica. Dicho de otro modo, se puede dar lugar a una oración que formalmente es idéntica a la que se desea generar siguiendo distintos senderos en la gramática. A lo largo de nuestro experimento esto sucedió en varias ocasiones, aunque aquí comentaremos el caso de la oración *There was a problem with the computer*. Aquí el participante *computer* podía ser especificado como un circunstancial de compañía o de instrumento. En ambos casos el SPL correspondiente generaba la oración deseada, al menos desde un punto de vista formal. No obstante, la utilización de distinta especificación semántica por parte de los alumnos llevó a un debate sobre qué opción era la más correcta desde el punto de vista semántico.
- e) Finalmente se dieron casos de errores por omisión que el alumno reconocía inmediatamente. En el caso de KPML, además, estos errores tenían consecuencias evidentes en el resultado generado, lo cual les movía inmediatamente a rectificar su análisis. Ejemplos de tales errores son olvidar que un sustantivo está en plural y no singular, que una oración está en tiempo pasado o polaridad negativa, o bien que un participante determinado es la persona a la que dirigimos el mensaje y no cualquier individuo en tercera persona. Esto constituye otro ejemplo de cómo el hecho de que el análisis oracional creado por el alumno dé lugar a un resultado tangible fomenta la reflexión sobre la lengua y permite a los alumnos detectar errores que son sólo producto del descuido, la falta de atención o interés.

5. RESUMEN Y CONCLUSIONES

La experiencia de integración de un generador automático, típicamente utilizado con fines investigadores o de desarrollo por diferentes grupos en Europa, como base para la enseñanza de la gramática funcional de la lengua en el aula virtual abre una serie de perspectivas innovadoras en el ámbito docente que no habían sido exploradas hasta el momento en nuestra Universidad.

Los beneficios de esta experiencia pueden resumirse en los siguientes puntos:

1. La creación de un tutorial en línea, distribuido en diferentes secciones como parte de una de las unidades del curso virtual de la asignatura de Lingüística Computacional permitió a los alumnos consultar la información necesaria para complementar las explicaciones y la demostración que se realizó durante las sesiones presenciales.
2. La creación de un archivo de ejercicios dirigidos a la exploración del funcionamiento de un generador automático, y disponible en línea en todo momento estimuló el aprendizaje activo y participativo de los estudiantes.
3. La integración de un generador automático como parte de la experiencia de aprendizaje ha supuesto un cambio muy significativo en la percepción de los aspectos funcionales de la lengua, como un mecanismo dinámico, más que como un conjunto estático de reglas.
4. El hecho de que los análisis oracionales de los alumnos sean probados mediante un programa informático supone un estímulo adicional, ya que el alumno recibe como premio una coincidencia plena entre la oración deseada y la generada por el programa.
5. Asimismo, las generaciones incorrectas llaman la atención del alumno sobre los detalles del análisis oracional y hacen evidente las diferencias a nivel estructural entre distintas oraciones de la lengua.

6. La corrección conjunta en clase de los ejercicios de generación estimuló los aspectos cooperativos y sociales que caracterizan el aprendizaje de calidad, aumentando el grado de implicación de los estudiantes en su propio aprendizaje, y agudizando su capacidad de análisis (véase Lavid, 2004).

AGRADECIMIENTOS

Agradecemos al Vicerrectorado de Innovación, Organización y Calidad de la Universidad Complutense de Madrid la concesión del proyecto titulado «*NTIC aplicadas al estudio funcional de la lengua: materiales para una docencia de calidad en el Campus Virtual*» que ha hecho posible la realización de este trabajo. También agradecemos a los alumnos de la clase de Lingüística Computacional (lengua inglesa) del presente curso su participación en esta experiencia.

BIBLIOGRAFÍA

- BATEMAN, J. A., MATTHIESSEN, C. M. (1991): *Text generation and systemic-functional linguistics: experiences from English and Japanese*, Pinter, Londres.
- BATEMAN, J. A. (1997): «Enabling technology for multilingual natural language generation: the KPML development environment». *Natural Language Engineering* 3 (1), p. 15-55.
- CAWSEY, A., BINSTED, K., JONES, R. (1995): «Personalised explanations for patient education». En: *Proceedings of the Fifth European Workshop on Natural Language Generation*. Leiden, the Netherlands. Faculty of Social and Behavioral Sciences, Rijks Universiteit, Leiden, pp. 59-74.
- COCH, J., DAVID, R., MAGNOLER, J. (1995): «Quality Test for a Mail Generation System». En *Proceedings of Linguistic Engineering 95*, Montpellier, Francia.
- DIMARCO, C., HIRST, G., WANNER, L., WILKINSON, J. (1995): «Healthdoc: customizing patient information and Health Education by medical condition and personal characteristics». En A. Cawsey (ed.) *Proceedings of the Workshop on Patient Education*. Universidad de Glasgow, Escocia.

- ELHADAD, M., ROBIN, J. (1997): «SURGE: a comprehensive plug-in syntactic realization component for text generation». *Technical Report*. Computer Science Dept. Ben Gurion University, Beer-Shiva, Israel.
- FAWCETT, R. P. (1990): «The COMMUNAL project: two years old and going well». *Network* 13/14, pp. 35-39.
- KITTREDGE, R., POLGUÈRE, A., GOLDBERG, E. (1986): «Synthesizing Weather Reports from Formatted Data». En: *Proceedings of the 11th International Conference on Computational Linguistics*. International Committee on Computational Linguistics. Bonn, Alemania, pp. 563-575.
- LAVID, J. (1995): «From Interpersonal Options to Thematic Realization in multilingual administrative forms». En: R. Kittredge (ed.) *Working Notes of the IJCAI-95 Workshop on Multilingual Text Generation*. Montreal, Canadá.
- LAVID, J. (1996): «Generating Thematic Choices for Multilingual Text Generation». En: Lindebjerg, E. Ore y O. Reigem (eds.) *Book of the ALLC-ACH '96*. Norwegian Computing Centre for the Humanities, p. 183-188.
- LAVID, J., ARÚS, J., ZAMORANO, J. R. (2002): «Developing a corpus-based generation grammar of Spanish: a contrastive study with English». En: Iglesias Rábade, L., y Doval Suárez, S. (eds.) *Studies in Contrastive Linguistics*. Servicio de publicaciones de la Universidad de Santiago de Compostela, p. 569-577.
- LAVID, J. (2003): «The Design of Contrastive (English-Spanish) Grammatical Resources or Multilingual Generation: the Grammar of Mood as an Example». Ponencia presentada en el Congreso *Computers, Literature and Philology 2002*. Albacete (UCLM).
- LAVID, J. (2004): «Lingüística inglesa y virtualización en un contexto presencial: reflexiones en torno a una docencia de calidad». *I Jornadas Campus Virtual UCM*, Editorial Complutense, Madrid.
- LAVID, J. (2005): *Lenguaje y Nuevas Tecnologías: nuevas perspectivas, métodos y herramientas para el lingüista del siglo XXI*. Ediciones Cátedra, Madrid.
- LAVID, J. (en prensa): «La generación del lenguaje en los sistemas de diálogo». En Llisterri, J. y Machuca, M., (eds.) *Los sistemas de diálogo*. Bellaterra, Servei de Publicacions de la Universitat Autònoma de Barcelona.
- MANN, W. C. (1983): «An overview of the PENMAN text generation system». En: *Proceedings of the National Conference on Artificial Intelligence AAAI*, pp. 261-265. También disponible como USC/Information Sciences Institute, RR-83-114.
- MANN, W. C., MATTHIESSEN, C. M. (1985): «Demonstration of the Nigel text generation computer program». En: Benson, J. D. y Greaves, W. S. (eds.) *Systemic Perspectives on Discourse*. Vol. 1. Norwood, NJ., Ablex, p. 50-83.
- MCDONALD, D. (1992): «Natural language generation». En: Stuart C. Sapiro (ed.) *Encyclopedia of Artificial Intelligence*, Nueva York, John Wiley and Sons, p. 983-997.
- NOT, E., STOCK, O. (1994): «Automatic generation of instructions for citizens in a multilingual community». En: *Proceedings of the European Language Engineering Convention*, París.
- PARIS, C., LINDEN, K. V., FISCHER, M., HARTLEY, A., PEMBERTON, L., POWER, R., SCOTT, D. (1995): «A Support tool for writing multilingual instructions». En: *Proceedings of International Joint Conference on Artificial Intelligence*. Montreal, Canadá.
- REITER, E. B. (1990): *Generating appropriate natural language object descriptions*. Tesis doctoral, Harvard University.
- REITER, E., MELLISH, C., LEVINE, J. (1995) «Automatic generation of technical documentation». *Applied Artificial Intelligence* 9.
- RÖSNER, D., STEDE, M. (1994): «Generating multilingual documents from a knowledge base: the techdoc project». En: *Proceedings of the 15th International Conference on Computational Linguistics*, vol. 1, Kyoto, Japón, p. 339-346.
- SPRINGER, S., BUTA, P., WOLF, T. (1991): «Automatic letter composition for customer service». En: R. Smith y C.Scott (eds.) *Innovative Applications of artificial intelligence 3*, AAAI Press, Menlo Park.
- ZAMORANO, J. R. (2002): *Exploration of the applications of multilingual generation grammar in language teaching*. Propuesta de tesis doctoral, Universidad de Bremen.